May 6th, 2003

The impact of non-uniform distribution of citations information on document relevancy [1].

Vasilii Shelkov

vasia@slac.stanford.edu

Abstract

In the large cross-referenced information archives or documents catalogs, the association of the document relevancy with a single number(rank) can be misleading. The non-uniform density of number of citations across the system, as well as correlations among the sources of these citations need to be taken into consideration. New ranking procedure, advocated in this note, provides a simple way to account for these, and other higher order effects.

[1] Patent is pending

Contents

List of Figures

1 Introduction

A great example of the complex and cross-referenced information system is World Wide Web(WWW). The idea of WWW came from CERN(European Organization for Nuclear Research), where it was first introduced by Tim Berners-Lee in 1990. Since then, the WWW index size has exceeded 3×10^9 pages, and continues to experience an explosive growth. To access the information from WWW, a whole family of so-called "search engines" has been introduced. One of the most recent advances in the quality of information retrieval came with the introduction of ranking method by Google Inc [1]. The main idea behind this approach was to combine pattern matching algorithm used to analyze page contents, and the page "popularity" index obtained from the global analysis of WWW interconnectivity. The chaotic structure of WWW development, and its sheer size give rise to a dual set of structural properties: on the one hand, WWW is a *interconnected graph*, on the other hand, it is a *pseudo-random statistical ensemble*. The original *PageRank*^{TM} addressed the graph part, but made no use of its randomness. If, for a given web-page, the distribution of the number of incoming links demonstrates some form of random statistical behavior(e.g. Gaussian-like shape), then there are ways to interpret its shape properties, and their relations to the effective ranking of this page. For example, for a pair of equally ranked web-documents, the narrower distribution in the number of incoming links per random sample of web-pages manifests a better, more uniform spread(over WWW) of its "referees", and should trigger some increase in its effective rank. At the same time, if, for a pair of web-pages, all their incoming links are coming from the same set of "referees"(i.e. 100% correlated in rank), their compound rank decreases and, becomes equal to the rank of a single page(i.e. pages become indistinguishable in terms of rank). If numbers of incoming links for a given page counted in samples of random web-pages are completely independent from each other and the sample the new ranking algorithm doesn't change the rank values. Thus, it is fair to say that the "one-number-per-page" ranking(*a la* Google) is a special("ideal") case of the new, more general page ranking technique described in this writeup [2]. The main goal of this note is to demonstrate that there is a simple, easy to implement way to account for most of these effects and arrive at the new sequence of page ranks.

2 Counting incoming links

2.1 Widths and tails

To demonstrate some features of the new ranking system, real web-sites of four well-known News Agencies are selected for the case study. Without revealing their actual names, let's call them simply **a.com, b.com, c.com** and **d.com**. For each of 150 independent samples of about $\sim$ 150K random web-pages each, we count the total numbers of quoted links pointing back to each of four web-pages under consideration. The distributions of the ratios of numbers of incoming links normalized to the total number of web-pages in each sample of random pages are shown in Figures 1 and 2. Plots show the "bell-curve" shaped distributions with well pronounced peaks and various degrees of asymmetries. The calculated means for **a,b, c.com**(marked by the arrows) are shifted away from the position of the most probable values(the peaks of distributions).

[2] Note, that everywhere in this note we use the mean number of incoming links as a measure of page relevancy(rank). In a way, this can be thought of as a zero-step in more sophisticated, iterative *PageRank*^{TM} algorithm used by Google

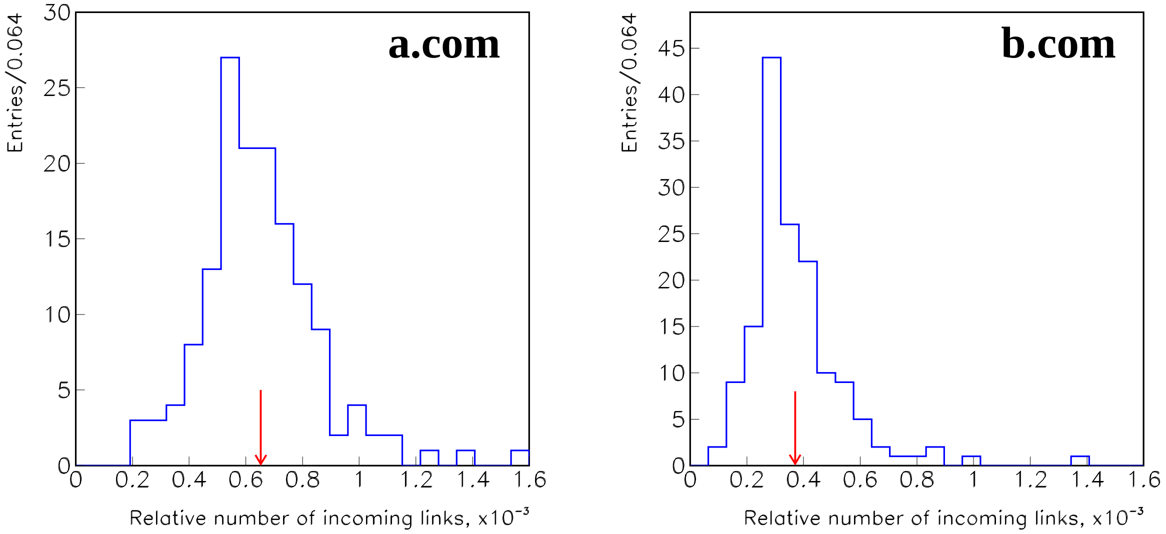

Figure 1: *Distributions of number of incoming links for a.com(left), b.com(right). Vertical arrow points to the mean value of the distribution.*

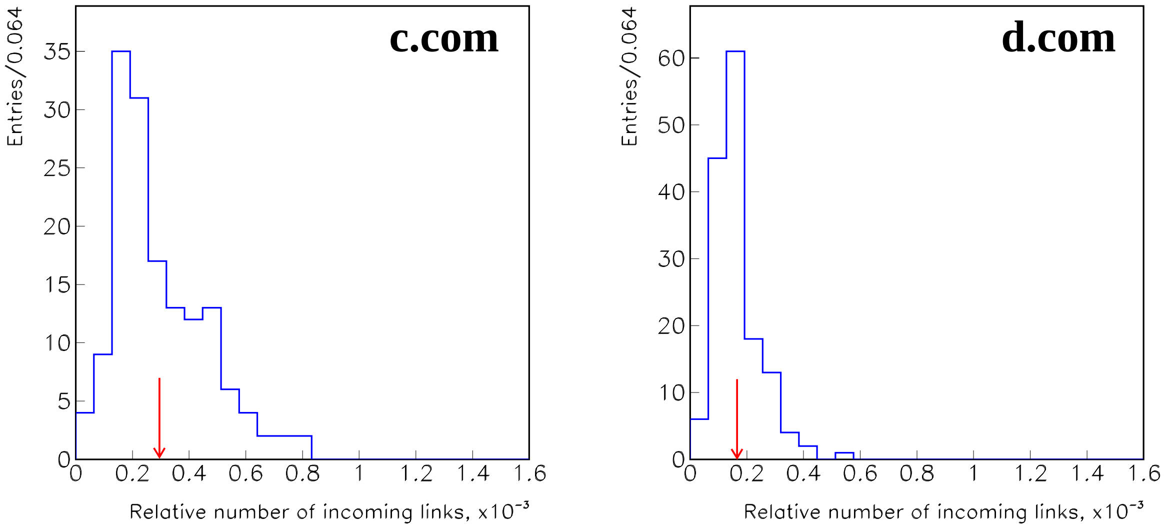

Figure 2: *Distributions of number of incoming links for c.com(left), d.com(right). Vertical arrow points to the mean value of the distribution.*

There could be a number of reasons behind various features of the distributions shapes. Large widths and tails may be caused by a clusters of "referees" voting either abnormally high or abnormally low, and thus indicate a "ranking controversy". Double peaked structures(see Figure 2, left) may indicate two separate regions of with different levels of popularity(e.g. US and Europe). Also, strange shapes and tails could indicate "unhealthy" activity of some companies to create islands of hyper-high voting activity(e.g. ask every customer to provide links back to them, even if they are not needed).

After a quick look at just 4 out of 3×10^9 web-documents, it becomes evident that the shapes of incoming links distributions can be complex, and it would be naive to characterize them with just a single number(i.e. mean document rank). To create a more sensible page ranking, some shape information has to be added into considerations.

2.2 Rank correlations

To compare rankings among various documents, one need to worry about correlations between sources of incoming links for these pages. In other words, we need to calculate the probability to find a random web-page containing links to **a.com** and **b.com** at the same time. The values of covariances and correlations can be calculate directly, according to their standard definitions:

$$Cov(x,y) = \sum_{i=1}^{Nsamples} (x_i - \overline{x})(y_i - \overline{y})$$

$$Cor(x,y) = Cov(x,y)/(\sigma_x \ \sigma_y)$$

$$\sigma^2 = \sum_{i=1}^{Nsamples} (x_i - \overline{x})^2$$

To calculate these sums, we take advantage of 150 samples of random web-pages used for making Figures 1 and 2. Each possible pair out of 4 pages has to be considered, and thus, the dimensions of covariance matrix are 4×4. The covariance and correlation matrices calculated for **a.com, b.com, c.com** and **d.com** are shown below(the overall factor of 10^{-6} in covariance matrix is omitted for simplicity):

$$Cov = \begin{vmatrix} 0.043 & 0.011 & 0.011 & 0.007 \\ 0.011 & 0.028 & 0.007 & 0.006 \\ 0.011 & 0.007 & 0.025 & 0.004 \\ 0.007 & 0.006 & 0.004 & 0.006 \end{vmatrix} \qquad Cor = \begin{vmatrix} 1.000 & 0.307 & 0.334 & 0.422 \\ 0.307 & 1.000 & 0.270 & 0.469 \\ 0.334 & 0.270 & 1.000 & 0.289 \\ 0.422 & 0.469 & 0.289 & 1.000 \end{vmatrix} \qquad (1)$$

One can immediately see a very significant level of correlations in general, and specifically between **b.com** and **d.com**(0.469). The values of each element of the correlation matrix are related to the fraction of random web-pages containing simultaneous references to a specific web-page pair(i.e. 0.334 ⇒ **a.com**-vs-**c.com**). To give some qualitative feeling to this very basic statistical phenomenon, the following simplified explanation can be offered. The main problem with comparing ranks of two highly correlated pages with a pair of loosely correlated pages, is that the "scale sizes" are different between two cases. The natural scales for ranks comparison can be derived from the widths of distributions shown in Figures 1 and 2. But in the limit of 100% correlations, the corresponding "scale size"(combined error on the rank difference) becomes equal to zero. The ratio of any observed difference in ranks with respect to such “zero scale” would go to infinity. Basically, one of two highly correlated pages becomes irrelevant(in page rank terms).

The combined impact of correlations on page ranking process can be evaluated by comparing sums of “on” and “off”-diagonal elements of correlation matrix(the one on the right shown in Equation (1)): $\sum_{On-diagonal} = 4$, $\sum_{Off-diagonal} = 4.2$. So, in this specific cases, the total impact of rank correlations is greater then the impact of ranks themselves.

3 New ranking algorithm

The method presented in this section can be viewed as an illustration on how to incorporate higher order effects into page relevancy determination. But first, we need to formulate what we are actually trying to achieve with this new algorithm.

We start by pre-selecting a sample of WWW web-pages which have a reasonable matching with the search pattern(i.e. a character string typed in the search engine window).

Then assume, that information we are searching for is distributed among the documents of this pre-selected sample, and the goal is to find a linear combination of these pages which would minimize this combination's resulting covariance S, and maximize its overall rank D.

In plain words, we are trying to find a way to re-rank pages to achieve the highest possible total relevancy(highest total rank), best overall quality(smallest possible total width), and the best diversification(least correlations):

$$D = \sum_{i=1}^{\text{matched}} \lambda_i d_i, \;\; d = w \cdot \overline{r}^2, \;\;\; S = \sum_{i=1}^{\text{matched}} \sum_{j=1}^{\text{matched}} \lambda_i \lambda_j s_{ij}, \;\;\; s_{ij} = \sum_{k=1}^{\text{N}-\text{random}} (r_i^k - \overline{r}_i)(r_j^k - \overline{r}_j)$$

where,
λ_i, λ_j - **new ranks** of web-pages i and j,
w - weight for text-pattern matching[3],
$\overline{r_i}$ - mean rank of page i,
r_i^k - rank of page i, for k-th sample of random web-pages,
“matched” - is a pre-selected sample of text-matched pages,
“N-random” - samples of randomly selected from the entire WWW web-pages.

To maximize $\frac{D}{S}$, we make first derivative equal to 0 [4].

$$\frac{\partial}{\partial \lambda}\left(\frac{D}{S}\right) = \frac{D}{S^2}\left(\frac{S}{D} \times \frac{\partial D}{\partial \lambda} - \frac{\partial S}{\partial \lambda}\right) = 0,$$

where,
$\lambda = \{\lambda_1, \lambda_2, ...\}$ - vector of new page ranks,
$\frac{D}{S}$ - is a constant factor for all λ_i's(can be dropped).

So, one could re-write and solve the following characteristic equation:

$$\frac{\partial S}{\partial \lambda} = \frac{\partial D}{\partial \lambda} \tag{2}$$

[3] Here, we don't discuss any pattern matching algorithms.

[4] Since $\left(\frac{S}{D} \times \frac{\partial D}{\partial \lambda} - \frac{\partial S}{\partial \lambda}\right) = 0$, it's easy to show that the second derivative for this expression is equal to $-\frac{D}{S^2}\frac{\partial^2 S}{\partial \lambda^2}$, and always less than 0 for positive elements of S and D.

The solutions of characteristic equation shown above comprise a new set of page weights.

Using mean values obtained from distributions shown in Figures 1 and 2 as page ranks, and importing covariance matrix from Equation (1), the characteristic equation for our case study can be written as follows:

$$\begin{vmatrix} 0.043 & 0.011 & 0.011 & 0.007 \\ 0.011 & 0.028 & 0.007 & 0.006 \\ 0.011 & 0.007 & 0.025 & 0.004 \\ 0.007 & 0.006 & 0.004 & 0.006 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \cdot w_1 \\ 0.37^2 \cdot w_2 \\ 0.30^2 \cdot w_3 \\ 0.17^2 \cdot w_4 \end{vmatrix} \tag{3}$$

where,
λ_i - unknown weights(new page ranks),
w_i - weights for text-pattern matching.

For the sake of simplicity, everywhere in the text below, we make all w_i equal to 1.

4 Practical examples

4.1 Ideal case of Poisson statistics

The probability of finding exactly x incoming links, in a given sample of random pages when the links are found independently of one another and page sample at an average rate μ per sample, is given by Poisson distribution function:

$$f(x;\mu) = \frac{e^{-\mu}\mu^x}{x!}, \text{ where } \overline{x} = \mu \text{ and } Var(x) = \mu$$

In the absence of correlations, substituting the mean numbers of links for each page as the diagonal elements of the covariance matrix, the characteristic equation (2) returns the mean numbers of links. So, *if the system is dominated by random statistical behavior, the proposed here method doesn't alter the existing ranking.* This behavior is important when pages with significantly different ranks are considered simultaneously. Given above, one should note that only non-random, intrinsic shape structures are going to change the values of input ranks(i.e. desired behavior).

$$\begin{vmatrix} 0.650 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.370 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.300 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.170 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \\ 0.37^2 \\ 0.30^2 \\ 0.17^2 \end{vmatrix} \Rightarrow \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65 \\ 0.37 \\ 0.30 \\ 0.17 \end{vmatrix} \tag{4}$$

4.2 Measured widths, no correlations

To make an example a bit more realistic, the measured from 150 samples covariances are placed as diagonal elements for the matrix:

$$\begin{vmatrix} 0.043 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.028 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.026 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.006 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \\ 0.37^2 \\ 0.30^2 \\ 0.17^2 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 \\ \lambda_2/\lambda_1 \\ \lambda_3/\lambda_1 \\ \lambda_4/\lambda_1 \end{vmatrix} = \begin{vmatrix} 1.00(1.00) \\ 0.49(0.57) \\ 0.34(0.45) \\ 0.42(0.25) \end{vmatrix} \tag{5}$$

The column of numbers in brackets show the ratios of input ranks. New ranking, in this case, accounts for a better shape of **d.com** links distribution (see Figure 2, right), and puts it above **c.com** which was favored initially. This is the first example of disagreement with the old ranking method.

4.3 Small correlations

Small(10%) correlations among all for pages are added to the covariance matrix described in Section 4.2:

$$\begin{vmatrix} 0.043 & 0.003 & 0.003 & 0.002 \\ 0.003 & 0.028 & 0.003 & 0.001 \\ 0.003 & 0.003 & 0.026 & 0.001 \\ 0.002 & 0.001 & 0.001 & 0.006 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \\ 0.37^2 \\ 0.30^2 \\ 0.17^2 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 \\ \lambda_2/\lambda_1 \\ \lambda_3/\lambda_1 \\ \lambda_4/\lambda_1 \end{vmatrix} = \begin{vmatrix} 1.00(1.00) \\ 0.37(0.57) \\ 0.18(0.45) \\ 0.07(0.25) \end{vmatrix} \tag{6}$$

Even with such low level of rank correlations, there is a significant drop in rank values for **c and d.com**.

4.4 Large correlations

Large(90%) correlations among all for pages are added to the covariance matrix described in Section 4.2:

$$\begin{vmatrix} 0.043 & 0.031 & 0.030 & 0.015 \\ 0.031 & 0.028 & 0.024 & 0.012 \\ 0.030 & 0.024 & 0.026 & 0.012 \\ 0.015 & 0.012 & 0.012 & 0.006 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \\ 0.37^2 \\ 0.30^2 \\ 0.17^2 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 \\ \lambda_2/\lambda_1 \\ \lambda_3/\lambda_1 \\ \lambda_4/\lambda_1 \end{vmatrix} = \begin{vmatrix} 1.00(1.00) \\ -0.1(0.57) \\ -0.4(0.45) \\ -1.0(0.25) \end{vmatrix} \tag{7}$$

With so much correlations, only **a.com** gets a positive rank. Three other pages have their ranks turn negative and need to be removed from the top list of search results(could be replaced with pages with lower ranks which didn't make the the top list initially). Filtering out the highly correlated pages with high ranks provides a new way for low ranking documents to make it to the top list.

4.5 Saving page #4

With the same matrix as in Section 4.4, we reduce correlations for **d.com** to just 10%:

$$\begin{vmatrix} 0.043 & 0.031 & 0.030 & 0.002 \\ 0.031 & 0.028 & 0.024 & 0.001 \\ 0.030 & 0.024 & 0.026 & 0.001 \\ 0.002 & 0.001 & 0.001 & 0.006 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \\ 0.37^2 \\ 0.30^2 \\ 0.17^2 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 \\ \lambda_2/\lambda_1 \\ \lambda_3/\lambda_1 \\ \lambda_4/\lambda_1 \end{vmatrix} = \begin{vmatrix} 1.00(1.00) \\ -0.4(0.57) \\ -0.8(0.45) \\ 0.1(0.25) \end{vmatrix} \tag{8}$$

A simple reduction of correlations for page **d.com** from 90% to 10% helps to recover its positive rank back and make it relevant.

4.6 Nominal case

Solving the system of equations with complete covariance matrix from Section 3 we get:

$$\begin{vmatrix} 0.043 & 0.011 & 0.011 & 0.007 \\ 0.011 & 0.028 & 0.007 & 0.006 \\ 0.011 & 0.007 & 0.025 & 0.004 \\ 0.007 & 0.006 & 0.004 & 0.006 \end{vmatrix} \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \\ 0.37^2 \\ 0.30^2 \\ 0.17^2 \end{vmatrix} \Rightarrow \begin{vmatrix} 1 \\ \lambda_2/\lambda_1 \\ \lambda_3/\lambda_1 \\ \lambda_4/\lambda_1 \end{vmatrix} = \begin{vmatrix} 1.00(1.00) \\ 0.30(0.57) \\ -0.1(0.45) \\ -1.0(0.25) \end{vmatrix} \qquad (9)$$

It turns out that **c.com** and **d.com** pages are not relevant if full correlation matrix is taken into consideration. This is an example of another source of bias in the old ranking system.

4.7 Racing for #1

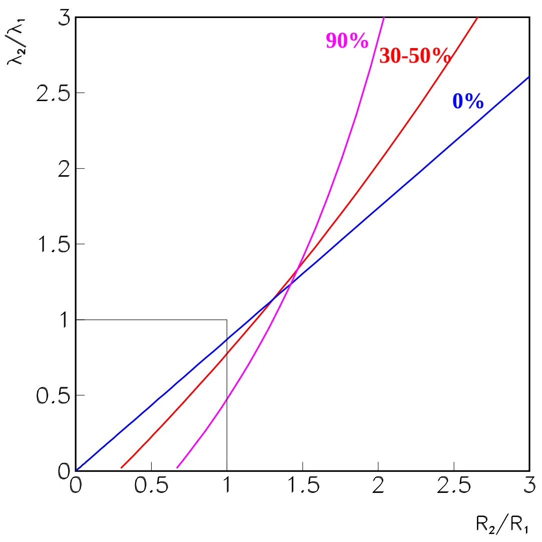

Figure 3: *Dependence of the output ratio of new ranks for pages 2 and 1, as a function of the input ratio of ranks. The percentage point next to lines indicate levels of correlations. Box represents values of ratios where first and second pages trade places.*

As shown in Figure 3, we plot a functional dependence between the output ratio of new ranks for pages 2 and 1 as a function of their input ratio(old ranks). It's evident from the plot, that the more correlation is put in the system, the larger gap between the values of two highest ranks(more space for lower ranks "to jump in").

4.8 Content correlations

In addition to correlations in ranks, the correlation between pages contents can be considered as well. If content covariance or correlations matrices are available through some algorithm, the techniques described in Section 3 can be used with no or minimal modifications. For example, one can count the number of common phrases, terms, article headings, and calculate a simplified correlation factor: $Corr = \sqrt{n/N}$, when N - total number of phrases, n - number of common phrases. The exact way of estimation levels of page content correlations is a very difficult task. Nevertheless, if possible, the accounting for content correlations can turn out to be the "gold mine" for improving search quality and diversifying search results.

5 Implementation model

At present, it seems like the best way to proceed is to apply the new ranking on the top of some existing ranking procedure. One can imagine doing this by offering rank and content(may be) filters working with the results returned by one of already existing search procedures. Since it doesn't seem practical to operate with 3×10^9 by 3×10^9 covariant matrix, one can imagine several ways of doing this. One way to reduce covariance matrix is to restrict calculations only to the upper layers of the rank ladder, or figure out the pairs of the most correlated sites and keep track of them only. More promising approach though, would be *to add to each web-document a list of 100-500 rank values calculated for unique sequence of samples of randomly chosen web-pages(500K-5000K). Then, not only page-to-page covariance, but also the distribution shape itself could be easily accessed and analyzed(see examples of distributions in Figures 1 and 2).* This could be useful if more sophisticated analysis will be applied at some future time. The procedure, described by Equation 2, can be made more flexible for getting the best performance in real-case scenario. For example, the Equation 1 can be re-written with an *extra damping parameter* α [5] as follows:

$$\left(\begin{vmatrix} 0.043 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.028 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.025 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.006 \end{vmatrix} + \alpha \cdot \begin{vmatrix} 0.000 & 0.011 & 0.011 & 0.007 \\ 0.011 & 0.000 & 0.007 & 0.006 \\ 0.011 & 0.007 & 0.000 & 0.004 \\ 0.007 & 0.006 & 0.004 & 0.000 \end{vmatrix} \right) \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{vmatrix} = \begin{vmatrix} 0.65^2 \cdot w_1 \\ 0.37^2 \cdot w_2 \\ 0.30^2 \cdot w_3 \\ 0.17^2 \cdot w_4 \end{vmatrix}$$

6 Conclusion

- for large and complicated systems(like WWW), it is hard to justify that a single number(rank) is an adequate way to assign page relevance,
- using real examples, we demonstrate various sources of biases resulting from not accounting for differences in shapes of distributions of incoming links, correlations among sources of incoming links,
- we propose to use statistical sampling to estimate the full[6] covariance matrix of WWW, and the density of incoming links distributions,
- we propose a new method for integration of second-order effects(mentioned above) into page ranking calculation at relatively small overhead cost,
- the importance of taking into account second-order effects into page relevance calculations will only grow with size of WWW, and will have to be addressed sooner or later,
- we believe that by using more detailed information about WWW structure, the new ranking method should provide a more comprehensive, and less biased way to assess page relevancy than the existing ranking techniques(the ones using mean rank or, in other words, just the diagonal elements of the WWW covariance matrix).

[5] Damping parameters for rank correlations can reflect the practical fact that even 100% rank-correlated pages can still be useful on average if their contents are very different.

[6] Applicable for pages with big enough number of incoming links to be treated through the sampling of random referee pages as described in the text.

References

[1] Sergey Brin and Lawrence Page,“The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Computer Science Department, Stanford University, Stanford, CA 94305, USA

A Selection of sample of random web-pages

In the absence of full access to the complete WWW index, one has to develop a way to obtain samples of random web-pages. In the absence of unified keys associated with pages, it's hard to imagine an easy way of doing this. The solution which seems intriguing is to search for random numbers in the page contents. The numbers are accepted universally around the World and have no language barriers. The main sources of random numbers are phone, fax numbers, zip codes, bytes sizes and many others. Below is the example of finding the relative number of incoming links for **x.com** as a function of number of digits in the random key(decimal number).

Key	Number of hits	Key and URL	Number of hits	Ratio($\times 10^{-3}$)
"517265"	170	"517265 x.com"	1	9.3
"45762"	4690	"45762 x.com"	35	7.5
"3876"	151000	"3876 x.com"	852	5.6
"495"	3880000	"495 x.com"	27100	7.0
"78"	28000000	"78 x.com"	187000	6.7

It is remarkable, that despite *5 orders(!)* of magnitude increase in the number of incoming links, the number of hits scales almost like a factor of 10, and the ratio stays very stable. The method is clearly a crude one, and ignores have many subtle issues like normalizing to the total number of links per page, excluding self-pointed, dead links and so on. Also, one might argue that it's not a truly random set of pages since there may be a whole class of pages with non-technical content which have strong deficit in numbers in their content. Samples of web-pages returned with 4-digit keys turn out to be practically independent from each other(perhaps due to the 7-digit structure of the phone numbers in US and abroad).

Nevertheless, this method seems to be a good enough tool to make "on the back of the envelope" type of calculations.